МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Ижевский государственный технический университет

имени М. Т. Калашникова»

Институт «Информатика и вычислительная техника»

Кафедра «Программное обеспечение»

Отчет

по лабораторной работе №2

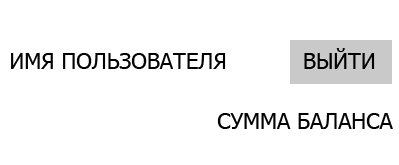
по дисциплине «Проектирование и конструирование программного обеспечения»

|  |  |  |
| --- | --- | --- |
| Выполнил  студент группы Б19-191-1: |  | Конычев А.К. |
|  |  |  |
| Принял  доцент: |  | Еланцев М.О. |

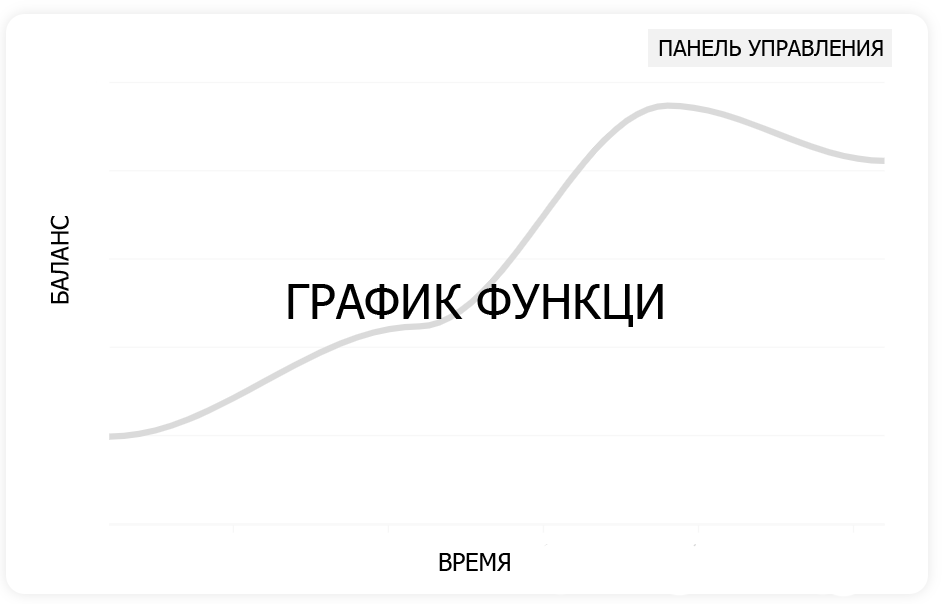
Ижевск

2022

**Разработка базового расписания.**

**Прототипы экранных форм.**

*Рисунок 1 – Отображение данных пользователя и выход из аккаунта.*



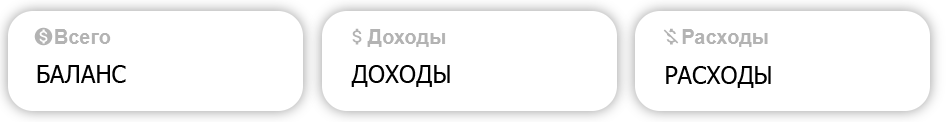
*Рисунок 2 – График баланса пользователя от времени*



*Рисунок 3 – Форма входа*

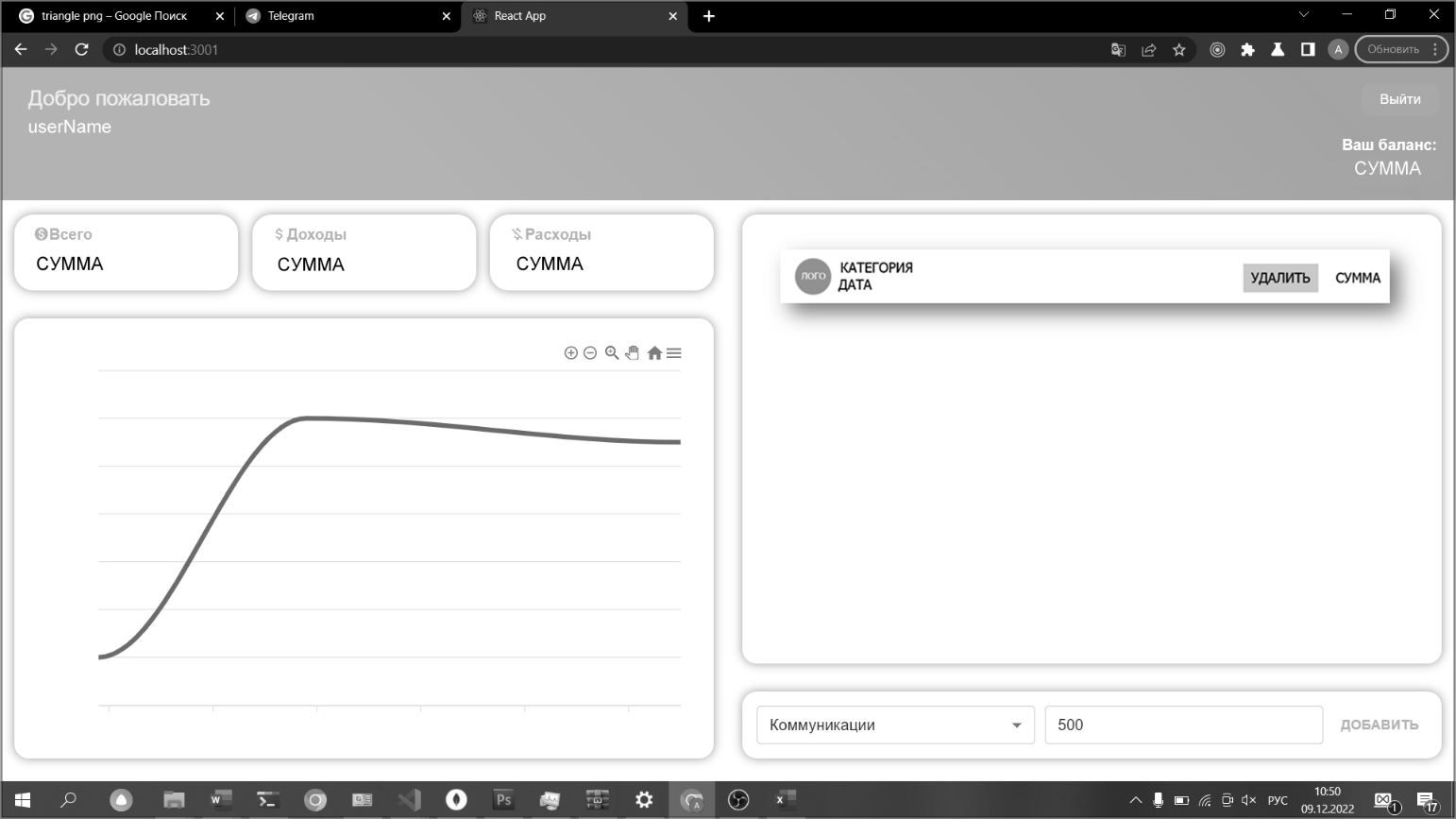


*Рисунок 4 – Форма создания дохода / расхода*

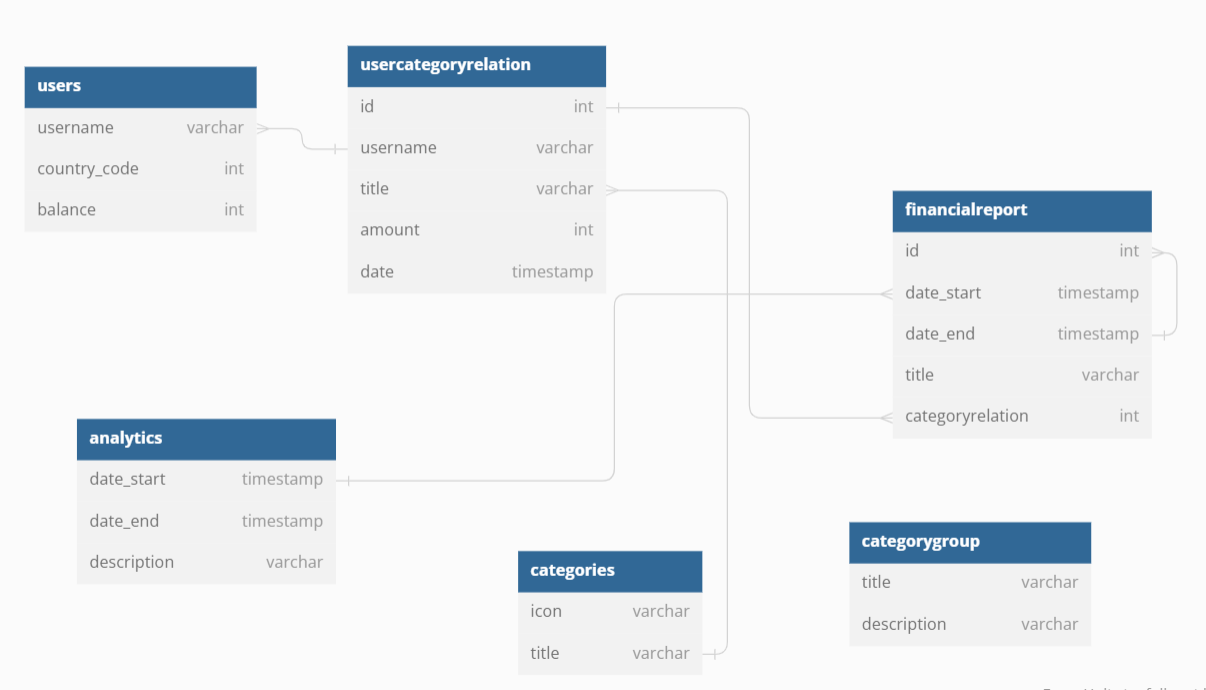


*Рисунок 5 – Текущая финансовая информация пользователя*

*Рисунок 6 – Информация о доходе / расходе*

**

*Рисунок 7 – Схема приложения*

**Диаграмма сущностей (ER).**

*Рисунок 7 – ER-диаграмма*

**Разработка API системы**

1. GetTransactions.

Получение списка транзакций пользователя. На вход подается токен авторизации пользователя, опционально дата начала выгрузки и дата окончания выгрузки. На выходе сервер отдает массив транзакций.

1. GetCategories.

Получение списка категорий расходов / доходов. На вход подается токен авторизации пользователя. На выходе сервер отдает массив категорий.

1. GetСategoryGroup.

Получение списка категорий расходов / доходов. На вход подается токен авторизации пользователя. На выходе сервер отдает массив описания определенной группы категорий.

1. PostTransaction.

Создание транзакции пользователя. На вход подается токен авторизации пользователя, сумма транзакции и ее категория. На выходные данные отдается ошибка, в случае если она возникла.

1. CreateUser.

Создание нового пользователя. На вход подается токен авторизации пользователя. В случае, если пользователь с данным никнеймом отсутствует, система создает новую запись пользователя в базе данных. На выходные данные отдается ошибка, в случае если она возникла.

1. GetAnalytics.

Получение аналитики по финансовому отчету. На вход подается токен авторизации пользователя и идентификатор финансового отчета. На выходе пользователь получает сформированную аналитику по указанному финансовому отчету.

1. CreateFinancialReport.

Создание финансового отчета пользователя. На вход подается токен авторизации пользователя, дата начала и окончания рассматриваемых транзакций. На выходные данные отдается ошибка, в случае если она возникла.

1. GetFinancialReport.

Получение финансового отчета пользователя. На вход подается токен авторизации пользователя и идентификатор финансового отчета. На выходе пользователь получает финансовый отчет, то есть сохраненную выгрузку транзакций за указанное время.

1. DeleteUser.

Удаление аккаунта пользователя, а также всех его транзакций, финансовых отчетов и аналитик. На вход подается токен авторизации пользователя. На выходные данные отдается ошибка, в случае если она возникла.

1. DeleteTransaction.

Удаление транзакции пользователя. На вход подается токен авторизации пользователя и идентификатор транзакции. На выходные данные отдается ошибка, в случае если она возникла.

**Иерархическая структура работы**

1. Разработка технического задания
   1. Сбор требований;
   2. Определение стадий и этапов разработки
      1. Определение стадий разработки;
      2. Определение сроков разработки;
   3. Общее описание
      1. Назначение продукта;
      2. Взаимодействие продукта;
      3. Допущения и ограничения продукта;
      4. Определение функций продукта;
2. Разработка приложения
   1. Backend-разработка
      1. Проектирование базы данных
         1. Определение структуры базы данных;
         2. Определение связей между сущностями;
         3. Определение взаимодействия с базой данных;
      2. Разработка API приложения
         1. GetTransactions;
         2. GetCategories;
         3. GetСategoryGroup;
         4. PostTransaction;
         5. CreateUser;
         6. GetAnalytics;
         7. CreateFinancialReport;
         8. GetFinancialReport;
         9. DeleteUser;
         10. DeleteTransaction;
      3. Сетевое взаимодействие
         1. Определение протокола взаимодействия;
         2. Обеспечение защищённости соединения;
         3. Определение местоположения хранения данных;
      4. Взаимодействие с UI
         1. Определение модели привязки данных;
   2. Frontend-разработка
      1. Дизайн
         1. Разработка макетов страниц
            1. Разработка главной страницы;
            2. Разработка страницы транзакций;
            3. Разработка карточки категории;
            4. Разработка формы создания транзакций;
            5. Разработка функционала списка транзакций;
            6. Разработка функционала списка категорий;
            7. Разработка страницы авторизации;
         2. Взаимодействие с backend
            1. Определение модели привязки данных;
            2. Настройка механизма обновления данных;
            3. Определение динамического взаимодействия с данными;
         3. Разработка общего стиль-кода приложения
            1. Создание логотипа приложения;
            2. Создание иконок элементов;
            3. Определение стиля и размера шрифтов;
3. Приемо-сдаточные испытания
   1. Подготовка и проведение демонстрации;
   2. Проведение испытаний;
4. Размещение приложения
   1. Аренда сервера;
   2. Развёртывание приложения;
   3. Размещение в магазине приложений;
5. Поддержка приложения
   1. Мониторинг работоспособности;
   2. Получение и обработка обратной связи;
   3. Улучшение работы приложения
   4. Добавление новой функциональности;

**Оценка времени работы по методу PERT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Работы | Количество | Оптимистичные трудозатраты | Пессимистичные трудозатраты | Наиболее вероятные трудозатраты |
| Реализация сущностей | 7 | 2 | 8 | 3 |
| Реализация форм | 7 | 30 | 60 | 50 |
| Реализация методов API | 10 | 40 | 70 | 55 |

Средние трудозатраты по каждой работе:

1. Создание сущностей = (8 + 4 × 3 + 2) / 6 = 3.6 чел.∗ час.
2. Создание макетов = (60 + 4 × 50 + 30) / 6 = 48.3 чел.∗ час.
3. Создание методов 𝐴𝑃𝐼 = (70 + 4 × 55 + 40) / 6 = 55 чел.∗ час.

Среднеквадратичная оценка:

1. Создание сущностей = (8 – 2) / 6 = 1 чел.∗ час.
2. Создание макетов = (60 – 30) / 6 = 5 чел.∗ час.
3. Создание методов 𝐴𝑃𝐼 = (70 – 40) / 6 = 5 чел.∗ час.

Еобщ. = 7 ∗ 3.6 + 7 ∗ 48.3 + 10 ∗ 55 = 913.3 чел.∗ час.

СКОобщ. = = 20.8 чел.∗ час.

Оценка суммарной трудоёмкости проекта с вероятностью 95%:

Е95% = 913.3 + 2 ∗ 20.8 = 954.9 чел.∗ час.